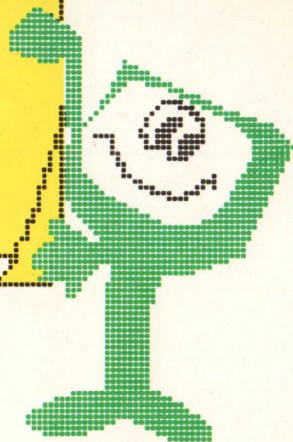
20 VIDEOLEZIONI DI BASIC
PER IMPARARE CON LO SPECTRUM

GRUPPO EDITORIALE JACKSON

6 Spectrum

16K/48K/PLUS

VIDEO BASIC SPECTRUM
Pubblicazione quattordicinale
edita dal Gruppo Editoriale Jackson
Direttore Responsabile:
Giampietro Zanga
Direttore e Coordinatore
Editoriale: Roberto Pancaldi
Autore: Softidea - Via Indipendenza 88 - Como
Redazione software:
Francesco Franceschini, Roberto Rossi,
Alberto Parodi, Luca Valnegri
Segretaria di Redazione:
Marta Menegardo
Progetto grafico:
Studio Nuovaidea - Via Longhi 16 - Milano
Impaginazione:
Silvana Corbelli
Illustrazioni:
Cinzia Ferrari, Silvano Scolari
Fotografie:
Marcello Longhini
Distribuzione: SODIP
Via Zuretti, 12 - Milano
Fotocomposizione: Lineacomp S.r.l.
Via Rosellini, 12 - Milano
Stampa: Grafika '78
Via Trieste, 20 - Pioltello (MI)
Direzione e Redazione:
Via Rosellini, 12 - 20124 Milano
Tel. 02/6880951/5

Autorizzazione alla pubblicazione Tribunale di Milano n° 422 del 22-9-1984
Spedizione in abbonamento postale Gruppo II/70 (autorizzazione della Direzione Provinciale delle PPTT di Milano).
Prezzo del fascicolo L. 8.000
Abbonamento comprensivo di 5 raccoglitori L. 165.000
I versamenti vanno indirizzati a: Gruppo Editoriale Jackson S.r.l. - Via Rosellini, 12 20124 Milano, mediante emissione di assegno bancario o cartolina vaglia oppure utilizzando il c.c.p. n° 11666203.
I numeri arretrati possono essere richiesti direttamente all'editore inviando L. 10.000 cdu. mediante assegno bancario o vaglia postale o francobolli.
Non vengono effettuate spedizioni contrassegno.

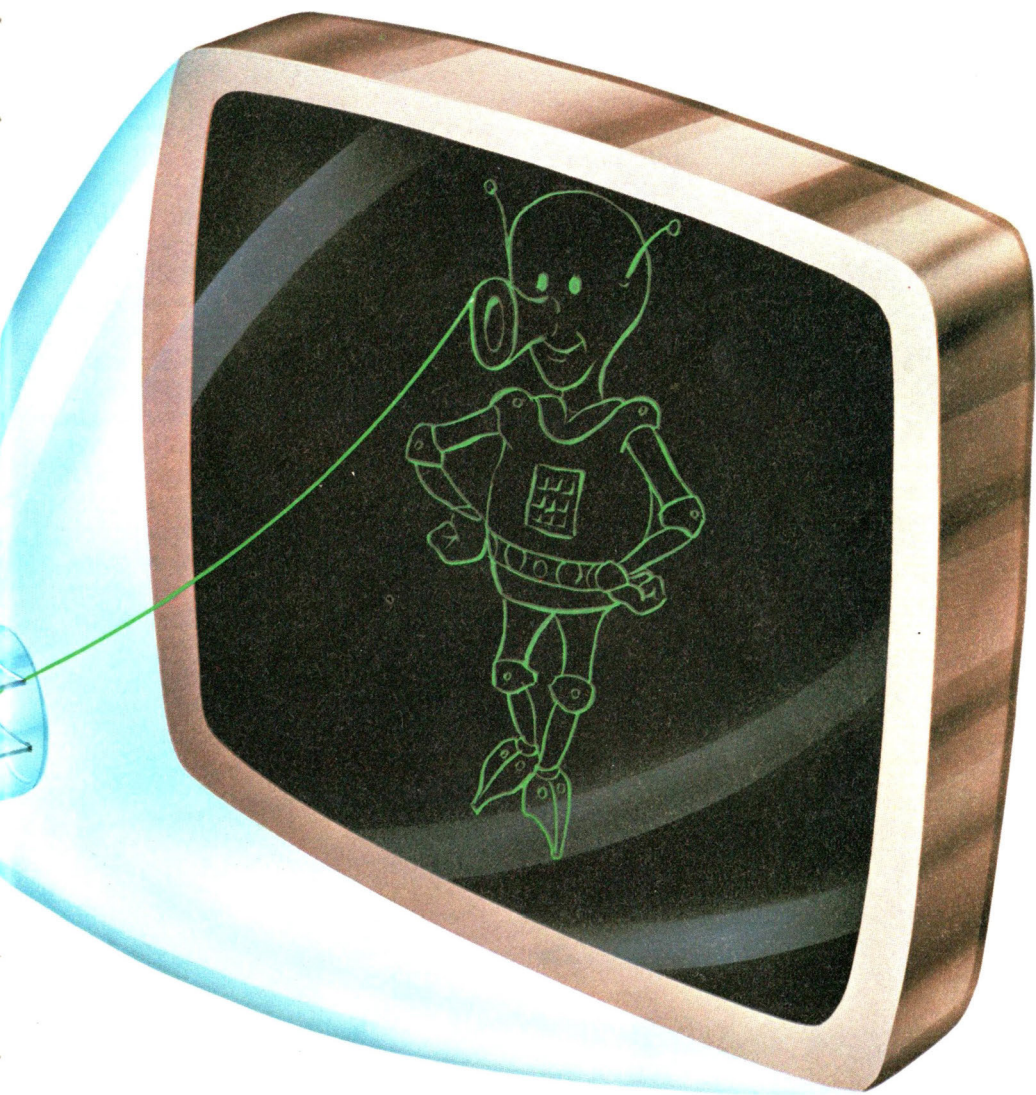
Gruppo Editoriale Jackson

SOMMARIO

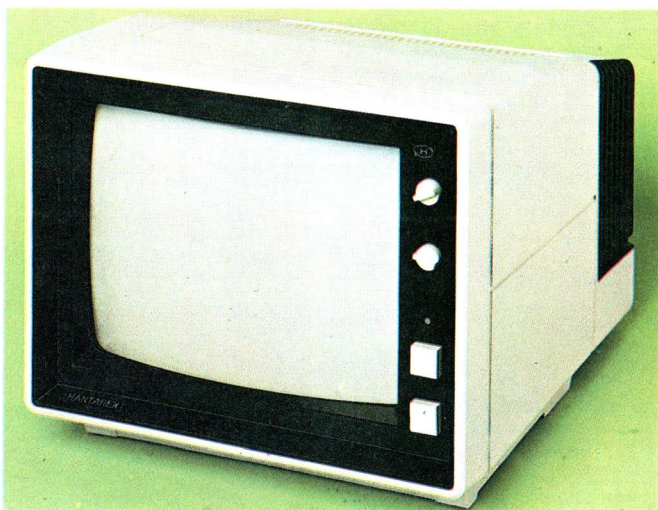
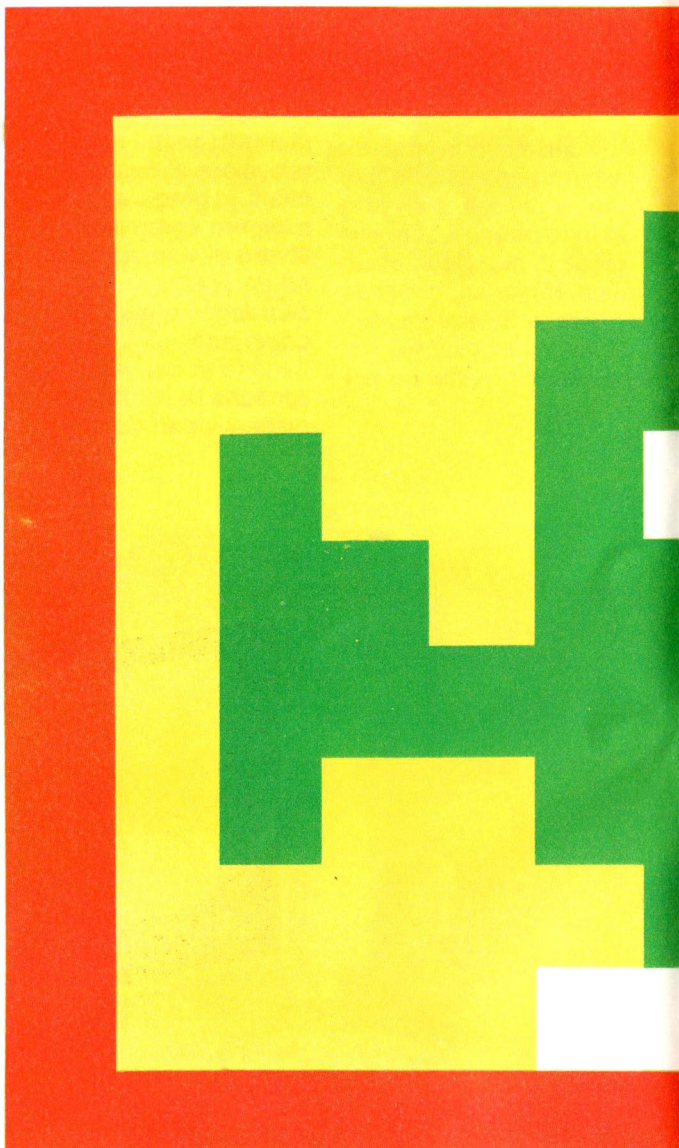
Introduzione

Il tuo computer è, tra le altre cose, anche una speciale stazione emittente televisiva. È infatti in grado di trasmettere via cavo le informazioni elaborate. Sotto questo aspetto, il programmatore diventa il "regista" dell'output dei dati trattati.
A lui sono, cioè, demandate le responsabilità di dare alle informazioni in uscita il massimo risalto e la maggiore chiarezza possibile.
Da qui l'importanza di conoscere sia l'hardware dedicato alla visualizzazione delle immagini, che le istruzioni e funzioni del BASIC capaci di controllare il formato e gli attributi.
Molto del successo dei tuoi programmi futuri dipende dalla familiarità con questi elementi.

Televisori e monitor

Il televisore ed il monitor - o, più generalmente, le unità video - costituiscono il principale dispositivo di uscita di un computer. Ad esse, infatti, viene normalmente affidato il compito di visualizzare tutte le informazioni, i dati ed i messaggi che costituiscono la base del fondamentale rapporto di interazione tra l'uomo e l'elaboratore.
Le funzioni svolte da un'unità video sono fondamentalmente tre:
1) visualizzare sullo schermo la maggior parte dei caratteri battuti sulla tastiera (tale funzione è chiamata eco);
2) visualizzare l'output dei programmi, consentendo di risparmiare tempo e carta quando non è necessaria una copia permanente;
3) inviare all'utente i messaggi dell'interprete BASIC (ad esempio le segnalazioni di errore).
L'uso delle unità video come dispositivi di output è abbastanza recente: fino a pochi anni fa le informazioni in uscita dal calcolatore erano infatti svolte quasi esclusivamente da stampanti e telescriventi. Ben presto, però, ci si rese conto che tali

HARDWARE

dispositivi erano assolutamente insufficienti per far fronte ad una mole di lavoro sempre crescente. Il loro costo di gestione, inoltre, era molto elevato (si rendevano necessarie montagne di carta e continue manutenzioni) ed avevano una capacità (ed una velocità) di visualizzazione abbastanza limitata rispetto alle esigenze dell'utenza.
Fu così che si pensò di affiancare alle pur necessarie stampanti delle unità di output più adeguate e flessibili all'uso di quanto non fossero quelle utilizzate fino a quel momento.
La scelta, come certamente avrai già immaginato, cadde sugli schermi video.
Essi rispondevano a tutti i requisiti richiesti: erano compatti, affidabili, economici (scarsa manutenzione e ridottissime spese di gestione), veloci.
Da allora, anno dopo anno, il loro utilizzo è diventato sempre più esteso ed intenso: ormai ai giorni nostri è impossibile riuscire a trovare un qualsiasi calcolatore che non sia provvisto di unità video.
Nei moderni personal le unità video che di solito vengono impiegate sono principalmente di due tipi: televisori e monitor.
Tra un televisore ed un

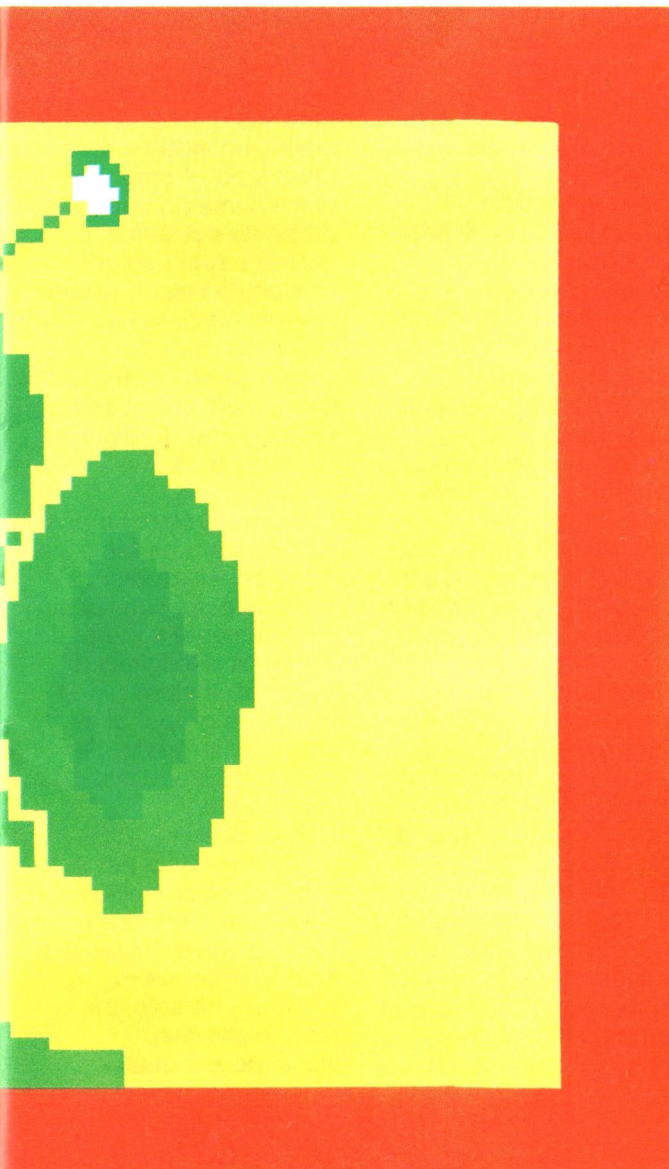
monitor non esiste, fisicamente e sostanzialmente, una grande differenza. Un monitor, infatti, non è altro che un televisore di ottima qualità al quale sono stati asportati tutti i circuiti adatti per la ricezione dei segnali attraverso l'antenna.
La qualità dell'immagine è ovviamente superiore a quella ottenibile da un normale televisore. Per un uso non professionale può essere comunque non necessario (se non inutile) ricorrere all'acquisto di un monitor: anche il televisore di casa è infatti in grado di svolgere egregiamente il lavoro di visualizzazione, ad un prezzo sicuramente più contenuto.
Il modo in cui vengono prodotte le scritte sul display video del tuo Spectrum è abbastanza semplice: esiste infatti una zona della memoria RAM nella quale vengono depositati - sotto forma di codice ASCII - tutti i caratteri che devono essere presentati sullo schermo.
A tale zona fa riferimento un apposito circuito, interno allo Spectrum, che "interfaccia" il calcolatore con l'unità video. Esso preleva cioè tutti i caratteri presenti nelle varie locazioni della memoria video e li invia al circuito del televisore, sotto forma di impulsi elettrici compatibili con il sistema o lo standard

televisivo.
Il televisore (od il monitor video) produce quindi una immagine visibile partendo da questi segnali elettrici.
Il modo in cui ciò avviene costituisce una delle più interessanti ed utili applicazioni della fisica elettronica e vede come componente principale un dispositivo chiamato cinescopio.
Un cinescopio (o tubo catodico) è un tubo a vuoto, cioè un contenitore di vetro nel quale è stato fatto il vuoto.
All'interno del tubo si

trova un "cannone elettronico" (basato su un filamento riscaldato dalla corrente che lo attraversa, come nelle lampadine) che produce un fascio (o pennello) di elettroni molto sottile. Gli elettroni godono di una singolare proprietà: quando colpiscono particolari sostanze fluorescenti - chiamate anche fosfori - fanno generare a queste sostanze una luminescenza, la cui durata può andare da alcuni millisecondi (millesimi di secondo) ad alcuni secondi, in funzione del tipo di fosforo e dell'intensità del raggio elettronico. L'immagine viene ricostruita proprio sfruttando questo fatto: il fascio di elettroni, "sparato" dal cannone e comandato da opportuni campi elettrici e magnetici, spostandosi da destra a sinistra e dall'alto verso il basso applica maggiore o minore intensità ai singoli punti di uno schermo che è stato ricoperto da un sottile strato di fosfori, provocandone una maggiore o minore luminosità.

La bassa risoluzione utilizza esclusivamente i caratteri (normali o grafici) presenti sulla tastiera.
Le immagini così ottenute non risultato molto dettagliate.

L'immagine viene pertanto ricostruita punto per punto dal pennello elettronico del tubo catodico alla stessa maniera in cui l'occhio di un uomo legge la pagina stampata di un giornale o di un libro. La velocità con cui tale pennello attraversa l'intero schermo è però talmente elevata da non poter essere minimamente percepita dall'osservatore umano: lo standard televisivo europeo prevede infatti che l'intero schermo, suddiviso in 625 righe orizzontali (mentre nello standard americano le

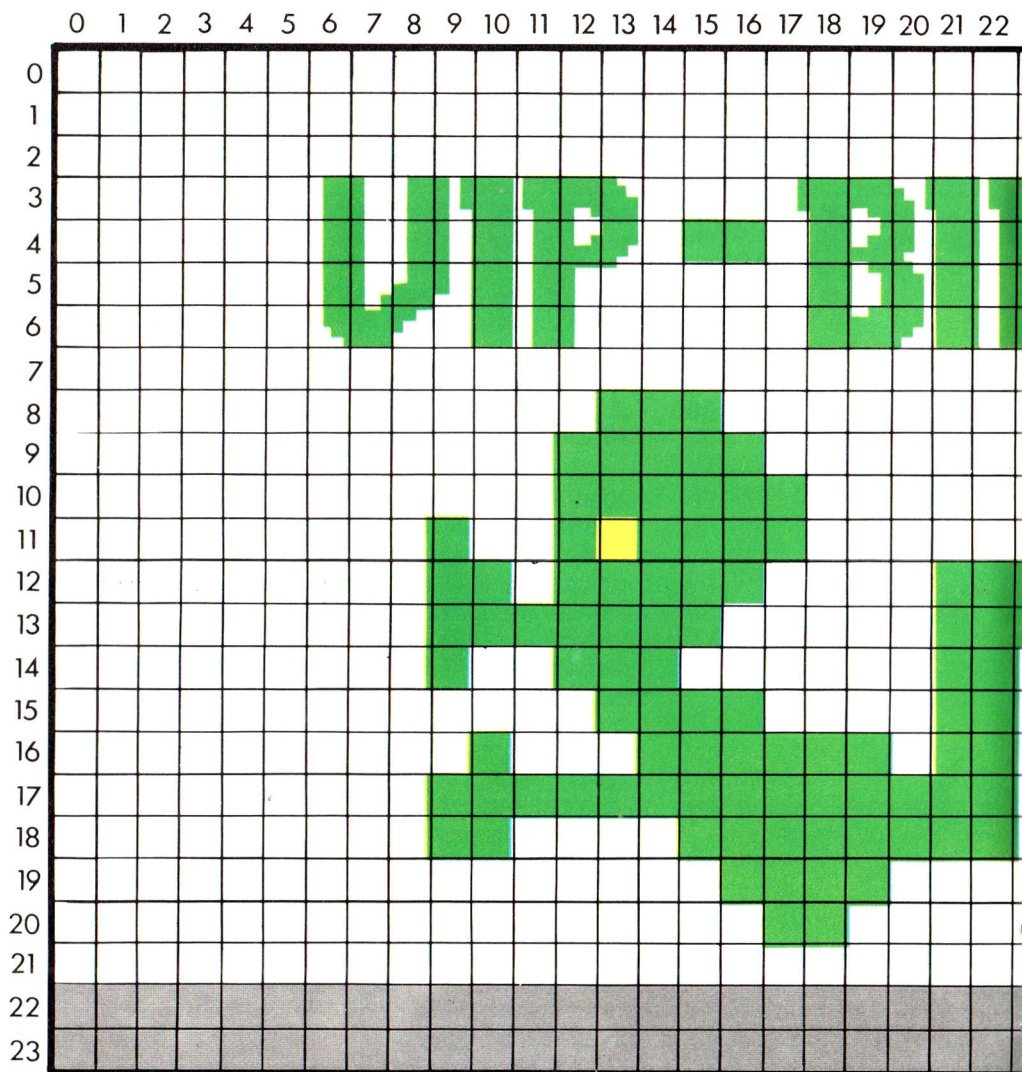
linee sono 525) sia percorso completamente ogni cinquantesimo di secondo. Il fenomeno della persistenza dell'immagine sulla retina fornisce quindi l'illusione di una immagine completa e simultanea.

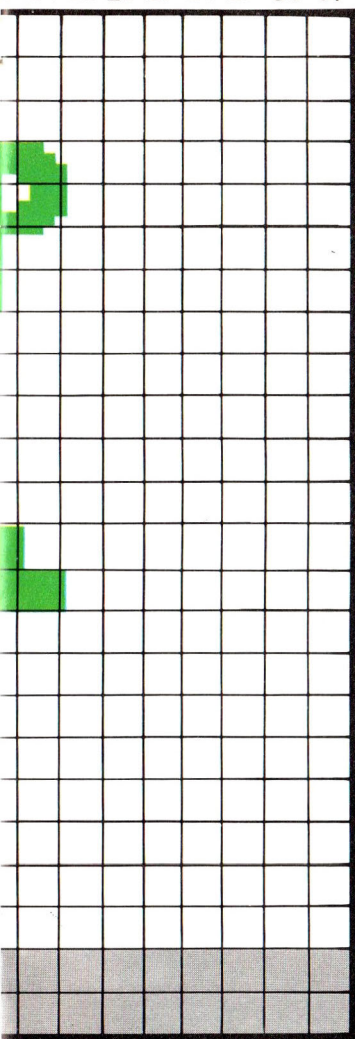
All'interno del televisore (o del monitor) vi sono, come accennato, anche dei circuiti destinati a comandare i movimenti del raggio elettronico sia in senso orizzontale che verticale.
Perché l'immagine sia visibile è però necessario che questi dispositivi lavorino simultaneamente.
Nel segnale video sono allora compresi, oltre alle informazioni riguardo all'intensità di ogni singolo punto dello schermo, anche appositi segnali destinati a coordinare il movimento del pennello elettronico. Sono i segnali di sincronismo.
Nel caso della trasmissione televisiva, tutte queste informazioni provengono da una telecamera; per i computer, invece, esiste un apposito circuito che, proprio come una telecamera, "legge"

Con l'alta risoluzione puoi indirizzare il singolo pixel determinandone o meno l'accensione.
Le immagini così ottenute risultano molto dettagliate.

l'immagine da visualizzare nelle varie locazioni riservate alla memoria video.
Il tipo di fosforo applicato sulla superficie dello schermo determina anche il colore del punto di collisione tra pennello elettronico e fosforo stesso.
Esistono in commercio diversi tipi di monitor: a fosfori verdi, gialli, ambra, ecc.. La scelta di un colore rispetto ad un altro è normalmente una questione di gusti e preferenze personali, anche se ultimamente giungono di continuo voci (e smentite) sul maggiore o minore affaticamento alla vista provocato da questo o quel tipo di fosforo.
Molto importante è invece la scelta delle dimensioni dello schermo. Un errore in cui spesso si incorre è infatti quello di credere che più grande sia lo schermo, migliore risulti la visione.
L'immagine, qualunque sia la dimensione del cinescopio, viene sempre suddivisa in 625 righe orizzontali: in uno schermo più piccolo le righe saranno quindi più sottili di quelle in uno schermo grande. La dimensione migliore dello schermo si avrà quando l'occhio umano, posto alla normale distanza di visione, non riesce più a distinguerne una dall'altra.
La dimensione in pollici dello schermo indica la lunghezza della diagonale, espressa appunto in pollici: personal computer e terminali video hanno solitamente schermi da 9" o 12" (9 o 12 pollici).
Un discorso a parte meritano gli schermi a colori.
Fermi restando i principi della televisione monocromatica, cioé ad un solo colore (spesso chiamato "in bianco e nero", anche se verde o giallo), un video a colori utilizza per il proprio funzionamento una proprietà dell'ottica, e cioé che tutti i colori esistenti possono essere ottenuti mediante la miscelazione e la combinazione di tre colori, detti per questo fondamentali: il rosso, il verde ed il blu.
All'interno del cinescopio, anziché uno solo, vi sono pertanto tre "cannoni elettronici". La superficie del cinescopio è interamente coperta da centinaia di migliaia di puntini di fosforo disposti a gruppi di tre, ciascuno in grado di emettere luce rossa, verde o blu.
Con lo stesso movimento visto per il video monocromatico (da destra a sinistra e dall'alto verso il basso) si muoveranno allora, anziché uno solo, tre raggi elettronici, ciascuno dei quali -

passando attraverso una maschera metallica opportunamente forata - è in grado di colpire soltanto i fosfori di un determinato colore.
Sulla superficie dello schermo si ottengono quindi tre immagini coincidenti, rispettivamente di colore rosso, verde e blu che combinandosi tra loro formano per l'occhio umano un'unica immagine a colori.
Questa tecnica si chiama sintesi additiva.

Schermo e memoria video

Abbiamo dunque visto il modo in cui le immagini vengono visualizzate sul tuo schermo televisivo.
Cerchiamo adesso di approfondire la relazione esistente tra ciò che si trova nella memoria del tuo Spectrum e quello che puoi vedere rappresentato sul video.
Esiste infatti una stretta dipendenza tra contenuto della memoria e formazione dell'immagine.
Il calcolatore, per produrre una immagine su uno schermo video, deve generare (come detto) un segnale simile a quello di una telecamera, in modo che il monitor (o il televisore) non noti alcuna differenza. I costruttori di computer, di conseguenza, sono stati costretti a ricorrere a stratagemmi per riuscire ad "ingannare" il video.
Ti ricordi quando, un paio di lezioni fa, parlammo della mappa della memoria dello Spectrum?
Bene, ora è arrivato il momento di rinfrescare l'argomento.
La memoria di un elaboratore non è a completa disposizione dell'utente: esistono infatti alcune zone (o aree) che non sono direttamente e liberamente utilizzabili per inserire dati ed istruzioni, ma che assolvono invece compiti, diciamo così, di ausilio e supporto.
A tali porzioni della memoria vengono quindi affidati, in sede di progetto e costruzione, incarichi non propriamente di "elaborazione", ma non per questo meno importanti ai fini del buon funzionamento del calcolatore. Così, alcune locazioni sono state dedicate a contenere i programmi, altre l'interprete BASIC, altre ancora - e sono quelle che ci interessano oggi - i caratteri da visualizzare sullo schermo.

Lo schermo dello Spectrum è diviso in due parti: la prima di 22 righe per 32 colonne disponibile per il Basic, la seconda di 2 righe per 32 colonne riservata al sistema operativo per l'input e i messaggi.

L'area della memoria di cui ci vogliamo occupare prende il nome di memoria video. Il suo scopo è quello di

contenere tutte le informazioni necessarie a costruire una immagine sul display televisivo.

Secondo te, sarà una memoria RAM od una memoria ROM? Non dovresti avere molti dubbi.
Per forza di cose, infatti, dal momento che vogliamo poter modificare il contenuto dello schermo, la memoria video deve appartenere alla zona di memoria RAM. In essa devono quindi avvenire cambiamenti tutte le volte in cui, per esempio, esegui una istruzione PRINT o INPUT, oppure (a parte casi particolari) batti qualcosa sulla tastiera che deve essere visualizzato sullo schermo.
Da sola, però, la memoria video non può fare molto: è infatti necessario che un apposito circuito video, chiamato di refresh (cioè "rinfresco") dell'immagine, legga - esattamente come una telecamera - tutta la memoria, carattere per carattere.
A quel punto il circuito video, venuto a conoscenza dei codici dei caratteri da visualizzare, consulta una particolare memoria ROM (chiamata anche generatore di caratteri) dalla quale preleva la "descrizione" grafica del carattere stesso. Come risultato finale si ottiene quindi un segnale video indicante se ciascun punto dello schermo deve essere acceso o spento.
Tutto ciò che compare sul video viene infatti rappresentato sotto forma di particolari combinazioni di puntini luminosi, chiamati pixel (abbreviazione dell'inglese picture element).
Ciascun puntino, potendo essere solo acceso o spento, è pertanto descrivibile da un singolo bit.
Senza addentrarci ulteriormente in dettagli tecnici, ti basti sapere che il circuito di refresh "prende" la combinazione di bit contenente la descrizione del carattere e la mette nella corrispondente casella dello schermo, accendendo o spegnendo i puntini come indicato precedentemente.
In totale i pixel luminosi disponibili sul tuo

Spectrum sono 49.152. Ad essi corrisponde nel complesso uno schermo composto da:
24 righe e 32 colonne, che formano in totale 24*32 = 768 caratteri.

Gli attributi e i colori

Lo schermo è quindi idealmente (ma anche fisicamente) divisibile in 768 (24 righe di 32 caratteri) posizioni dove i caratteri possono essere stampati. Ognuno di essi è rappresentato quindi da un quadrato di punti con dimensione 8 * 8.
Ciascuno di questi caratteri è inoltre rappresentabile in vari modi, per esempio: fisso, lampeggiante o colorato.
Il colore, la luminosità ed il lampeggio rappresentano, per un carattere visualizzato in una certa posizione dello schermo, i cosiddetti attributi. Così, quando stampi qualcosa sul video non fai altro che modificare la combinazione di alcuni degli attributi assegnati in precedenza a quella posizione.
Normalmente, cioè quando non imponi particolari specifiche - per esempio, riguardo il colore -, l'unico attributo che subisce modifiche è soltanto quello che riguarda l'accensione o lo spegnimento dei pixel che devono comporre il carattere.
Tutti gli altri attributi restano inalterati.
Esiste comunque la possibilità di variare a piacimento, per mezzo di particolari comandi, tutti gli attributi.
Vediamo per prima cosa il colore.
Ecco la lista dei colori disponibili sul tuo Spectrum, nell'ordine in cui sono riportati sui tasti numerici:

---

0 - nero
1 - blu
2 - rosso
3 - porpora, o magenta
4 - verde
5 - blu chiaro, o ciano
6 - giallo
7 - bianco

In un televisore in bianco e nero i numeri corrispondono ad una diversa gradazione di grigio, in ordine dal più chiaro al più scuro.
Ogni carattere è associato a due colori: il colore dell'inchiostro, o colore di stampa, ed il colore della carta, o colore dello sfondo.
Il colore iniziale dell'inchiostro, all'atto dell'accensione, è il nero, quello della carta è il bianco.
Con i comandi PAPER n (carta) e INK n

(inchiostro) è però possibile cambiare i colori. Per vederne l'effetto prova ad eseguire il seguente programma (per terminare sarà sufficiente che tu prema un tasto qualsiasi):

```
 10 FOR I=0 TO 7
 20 PAPER I
 30 FOR J=0 TO 7
 40 INK J
 50 PRINT "PROVA DI STAMPA"
 60 PAUSE 50
 70 IF INKEY$ <>" " THEN GO TO 110
 80 NEXT J
 90 NEXT I
100 GO TO 10
110 REM FINE
```

I parametri di PAPER e INK (n) possono assumere pertanto tutti i valori compresi tra 0 e 7. È però possibile assegnare anche i valori 8 e 9, nonostante che a questi numeri non corrisponda alcun colore.
Il parametro 8 significa "trasparente", cioè non altera l'attributo della posizione quando viene stampato il carattere; tutto resta quindi come specificato in precedenza.
9 significa invece "contrasto". Il colore dell'inchiostro o della carta, a seconda del comando che avrai usato, viene fatto contrastare con l'altro, diventando bianco su un colore scuro (nero, blu, rosso, magenta) e nero contro un colore chiaro (verde, ciano, giallo o bianco).
Oltre alla carta ed all'inchiostro è possibile far assumere anche alla cornice dello schermo uno qualunque dei colori disponibili.
Il comando che effettua questa operazione è BORDER n (bordo).
Il bordo può assumere uno qualunque degli otto colori normali (non 8 e 9). Questo semplice programma te ne illustrerà il funzionamento:

```
10 FOR I=0 TO 7
20 BORDER I
30 PAUSE 50
40 NEXT I
50 REM FINE
```

Gli ultimi due comandi che controllano gli attributi sono BRIGHT n (luminosità) e FLASH n (lampeggio).
BRIGHT n determina la luminosità dei caratteri stampati dal momento dell'esecuzione del comando in poi: n=0 per luminosità normale, n=1 per luminosità extra e n=8 per trasparenza.
Se n non è 0, 1 o 8, si ha la segnalazione che il numero specificato non è un valore consentito:

sullo schermo compare infatti INVALID COLOUR (colore non valido).
FLASH n definisce invece se i caratteri stampati successivamente al comando devono essere lampeggianti o stabili; n=0 per stabili, n=1 per lampeggianti, n=8 per nessun cambiamento. Anche qui, se n non è un numero valido, si ha una segnalazione di errore.
Vi sono poi altri comandi INVERSE e OVER che non controllano gli attributi, ma agiscono sulla matrice di punti stampata sullo schermo. I parametri di cui fanno uso sono soltanto 0 e 1, esattamente come FLASH e BRIGHT. Come avrai intuito, se introduci il comando INVERSE 1, il tuo Spectrum stamperà, da quel momento, i caratteri in negativo. Con INVERSE 0 porterai tutto alla normalità.
Il comando OVER 1 attiva, invece, un particolare tipo di sovraimpressione. Normalmente, infatti, quando stampi un carattere in una qualsiasi posizione dello schermo il simbolo che vi era eventualmente rappresentato viene cancellato e sostituito dall'ultimo. Con OVER 1, invece, il nuovo carattere viene sommato, ossia sovrapposto, a quello precedente. Ciò ti permette, ad esempio, di sottolineare un carattere. Prova prima con:

```
10 PRINT "A"; CHR$(8); : REM stampare il carattere di codice 8
   equivale a tornare indietro col cursore.
20 PRINT "_": REM il simbolo _ si ottiene premendo SYMBOL SHIFT
   e 0.
```

Poi con

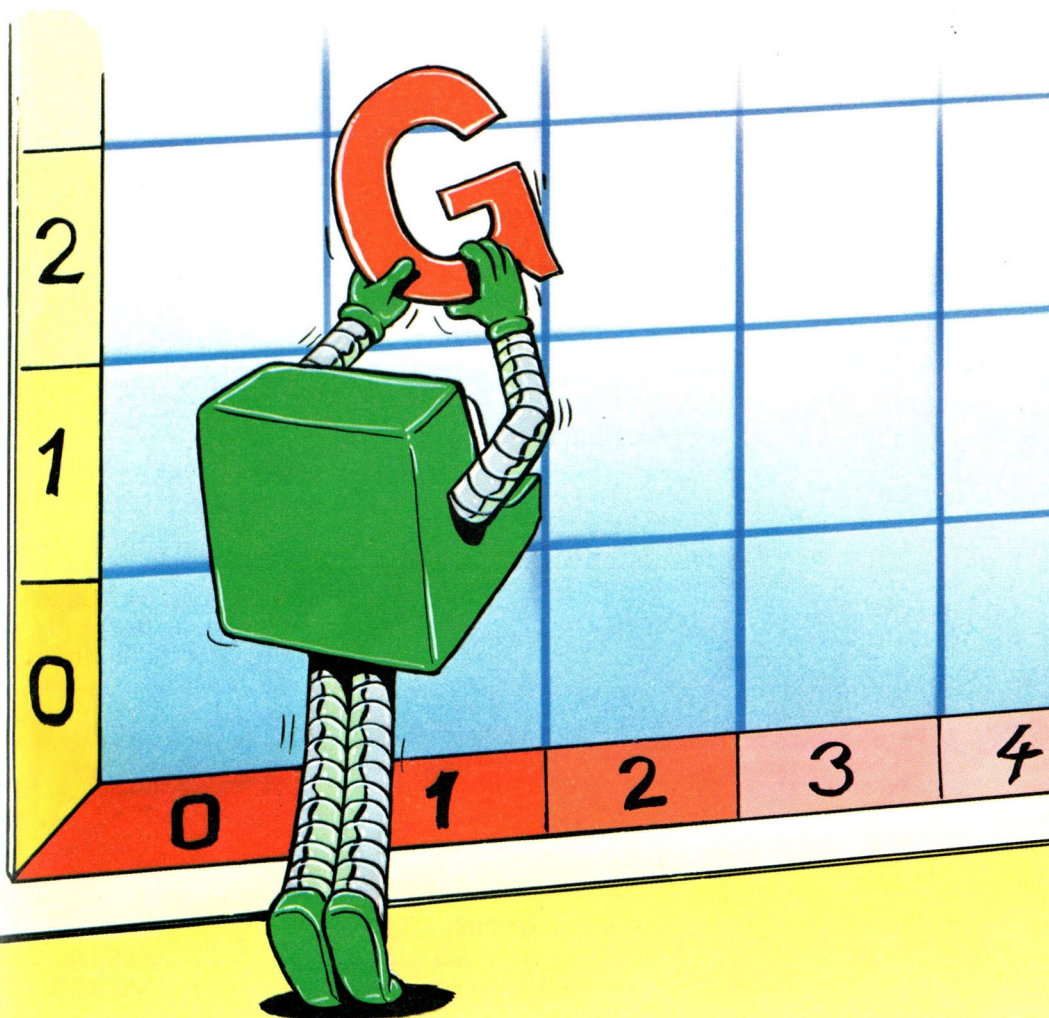
```
10 PRINT "A"; CHR$(8);
20 PRINT OVER 1; "_"
```

visto la differenza?!!

Funzioni di controllo della stampa su video

Nell'ambito delle istruzioni di visualizzazione previste dal BASIC, particolare importanza rivestono tutti i comandi che controllano e modificano, a piacere del programmatore, la posizione del cursore e - di conseguenza - delle scritte sullo schermo.
Abbiamo finora adoperato l'istruzione PRINT in numerose occasioni, utilizzandola per visualizzare tutti i risultati, i messaggi e le scritte che di volta in volta ci sono stati convenienti o necessari.
Ciò che però ancora ci manca è la capacità di controllare completamente questa istruzione, consentendoci per esempio di ottenere in uscita i risultati disposti in una certa posizione dello schermo, oppure ordinati ed incolonnati in un formato non necessariamente impostoci dal nostro calcolatore.
Detto in una parola (peraltro molto usata nel ramo dell'informatica), vogliamo saper formattare le scritte sullo schermo.
Questo è pertanto l'obiettivo delle istruzioni che oggi ci proponiamo di imparare.
La cosa importante da sottolineare è che tutte queste istruzioni non indicano al calcolatore cosa stampare, ma soltanto DOVE stampare.
Capito la differenza?
Vediamole adesso una per una, insieme a qualche esempio esplicativo.

AT

Scopo della funzione AT è quello di spostare la posizione di stampa (cioè il punto dello schermo dove sarà stampato il prossimo carattere) alla linea ed alla colonna specificata. Abbiamo già visto come lo schermo del tuo Spectrum dedicato al Basic sia stato diviso in 22 righe e 32 colonne. Le linee sono state

numerate da 0 a 21 (dall'alto in basso) e le colonne da 0 a 31 (da sinistra a destra).
La linea e la colonna che specificherai insieme al comando AT dovranno pertanto essere comprese in tali intervalli, pena l'invio del messaggio di errore INTEGER OUT OF RANGE (fuori dallo schermo).
Un'ultima osservazione. Avrai certamente intuito come la funzione AT, e lo vedremo anche più avanti nella parte della lezione dedicata alla programmazione, possa essere estremamente utile in numerose occasioni.
Occorre però prestare attenzione al fatto che con essa vengono ad essere alterati i normali meccanismi di scrittura sullo schermo. Può perciò darsi che un comando AT, impartito magari senza averci pensato troppo, porti a scrivere SOPRA una scritta già presente sullo schermo, provocandone logicamente la cancellazione.
Questo fatto, che peraltro in talune occasioni può anche essere utilmente sfruttato, va sempre ben tenuto in considerazione per evitare l'insorgere di simili, spiacevoli e fastidiosi inconvenienti.

Esempi

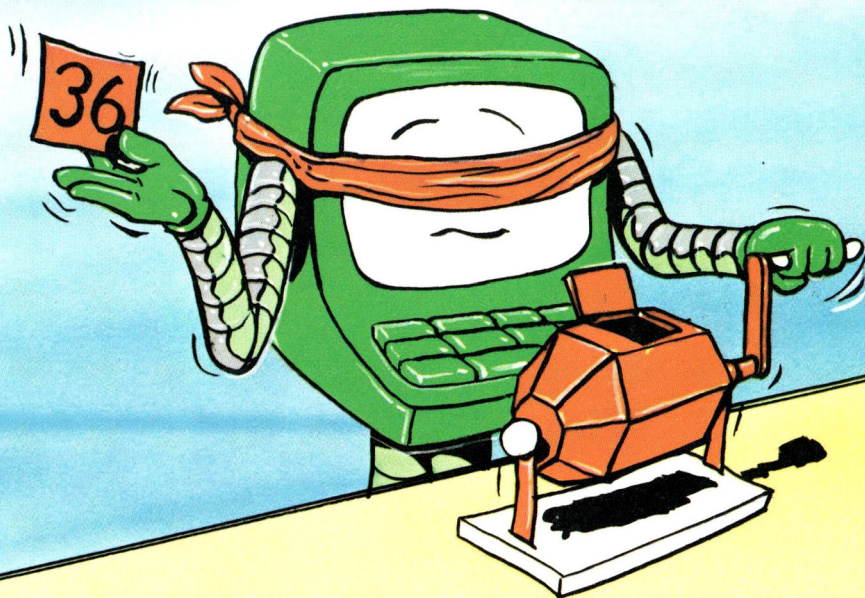
PRINT AT 10, 15; "MICIO"	stampa la parola "Micio" esattamente al centro dello schermo.
PRINT AT 21, 27; "GATTO"	stampa la parola "Gatto" nell'angolo in basso a destra dello schermo.
PRINT AT 0,27; "GATTO"	stavolta il felino viene visualizzato nell'angolo in alto a destra.

Sintassi della funzione

AT linea, colonna

dove linea deve essere un numero compreso tra 0 e 21 e colonna tra 0 e 31.

RND

Non è raro trovare casi in cui occorra, nell'ambito di un programma, disporre di numeri casuali.
Potresti, ad esempio, aver bisogno di un algoritmo che simuli l'estrazione a sorte di un numero, o il lancio dei dadi o ancora l'uscita delle carte da gioco dal mazzo.
La funzione RND ti risolve egregiamente questo tipo di problemi: crea, infatti, dei numeri casuali (random in inglese) compresi tra 0 e 1 ($0 <= R < 1$).
Il tuo Spectrum produce una sequenza di numeri casuali eseguendo dei calcoli su di un numero di partenza chiamato seme, generato al momento dell'accensione.
Poiché i numeri generati da RND sono il risultato di una complessa serie di calcoli, è più corretto

parlare di numeri pseudocasuali.
Diversamente dalle altre funzioni, RND non necessita di un argomento: PRINT RND è pertanto una istruzione corretta e determina la stampa di un numero casuale.
Prova ad introdurre e a far girare il seguente programma, per renderti conto del suo funzionamento:

```
10 FOR C = 1 TO 100
20 PRINT C; "   "; RND,
30 NEXT C
```

Poiché la gamma dei numeri compresi tra 0 ed 1 non è adeguata alla maggior parte delle applicazioni, usa la seguente formula per stabilire tu stesso il "range" entro il quale dovranno essere prodotti i numeri casuali:

```
LET R = INT ((LS - LI + 1) * RND) + LI
```

Dove R sta per numero Random, LS è il Limite Superiore del "range" ed LI il Limite Inferiore.
Per numeri Random interi compresi tra 9 e 18, applicando la formula avrai:

```
LET R = INT ((18 - 9 + 1) * RND) + 9
```

Per numeri compresi tra 0 e 10, sarà:

```
LET R = INT ((10 - 0 + 1) * RND) + 0
```

cioè

```
LET R = INT (11 * RND)
```

Sintassi della funzione

RND

RANDOMIZE

Il comando RANDOMIZE è usato soprattutto per determinare da quale punto della sequenza la funzione RND debba iniziare ad estrarre i numeri casuali.
Prova con:

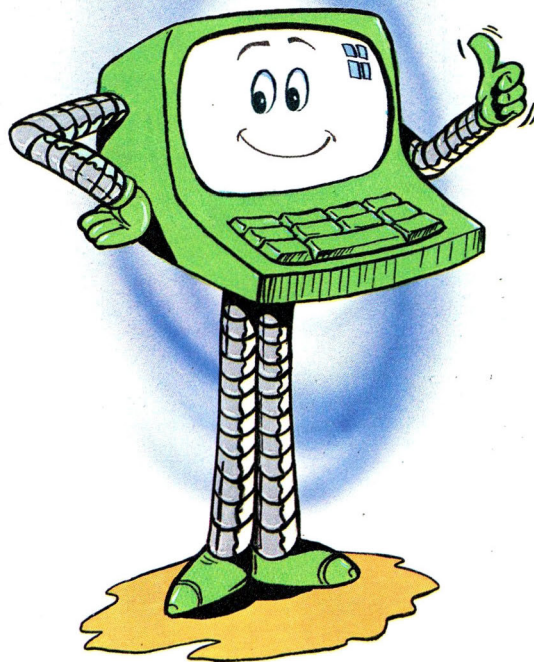
```
10 RANDOMIZE 22
20 FOR C = 1 TO 10
30 PRINT RND
40 NEXT C
```

Annota i 10 numeri casuali, poi fa partire nuovamente il programma con RUN e confrontali.
Modifica adesso nella linea 10 il numero dopo RANDOMIZE e riprova ...
... Visto la differenza?
Cambia nuovamente la linea 10.
Scrivi:

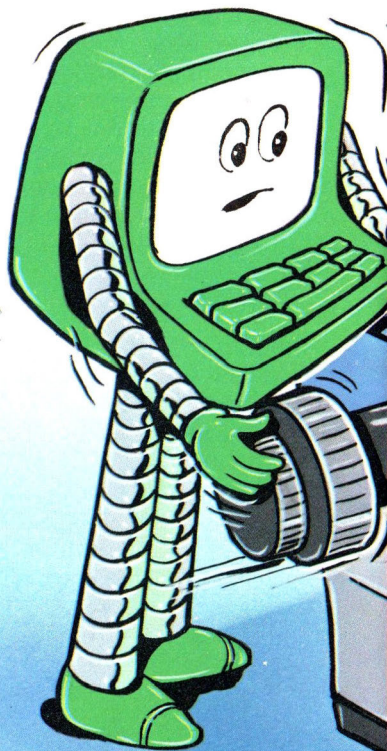
```
10 RANDOMIZE
```

oppure

```
10 RANDOMIZE 0
```

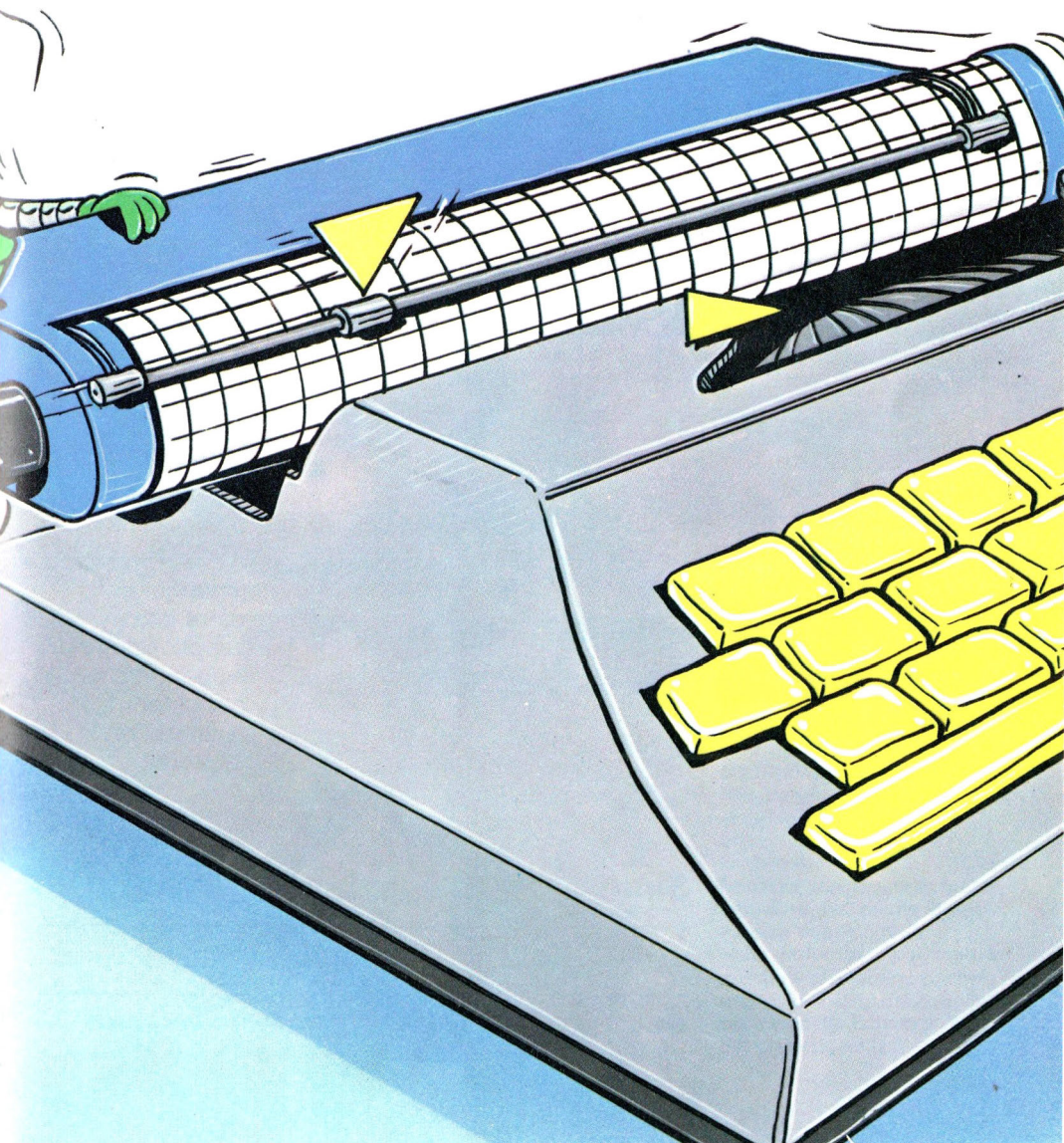
lasciando invariate le altre linee.

Fa nuovamente girare il programma più volte, annotando e confrontando le varie serie di numeri: non ne troverai più una identica all'altra.
In conclusione:

- Se vuoi generare delle serie di numeri casuali sempre uguali tra loro, allora fa precedere alla funzione RND il comando RANDOMIZE seguito sempre dal medesimo numero.
- Se vuoi invece ottenere sequenze di numeri casuali sempre diverse, precedi la funzione RND con RANDOMIZE 0 o semplicemente RANDOMIZE.

Sintassi del comando

RANDOMIZE [numero]

TAB

Come AT, anche la funzione TAB() si usa solo all'interno delle PRINT: essa opera infatti come la tabulazione di una normale macchina da scrivere.

La funzione TAB() viene normalmente utilizzata per allineare le stampe in colonne verticali, incolonnandole in punti prestabiliti.
Supponi infatti di dover

visualizzare alcuni dati sullo schermo del tuo video, allineati in un certo ordine. Grazie a TAB() puoi evitare i noiosi (e talvolta complicati) calcoli per incolonnare esattamente tutti gli elementi.

TAB () rimane sulla stessa linea dove si trova il cursore, a meno che la posizione di stampa specificata non sia precedente alla posizione attuale.
In tal caso si sposta alla linea successiva.

Esempi

```
PRINT NOME$; TAB(13); COGNOME$
```

Con questa istruzione, indipendentemente dalla lunghezza della stringa contenuta in NOME$, si avrà la stampa delle due variabili con questa disposizione: la prima partendo dalla colonna 0, la seconda dalla colonna 13.

```
10 PRINT TAB(2); "NUMERO"; TAB(12);
   "QUADRATO"
20 FOR I=1 TO 15
30 PRINT TAB(4);I;TAB(15);I↑2
40 NEXT I
```

Il programma qui a fianco mostra una semplice applicazione di TAB: stampa infatti, allineati per colonna, i primi 15 numeri con i rispettivi quadrati.

Sintassi della funzione

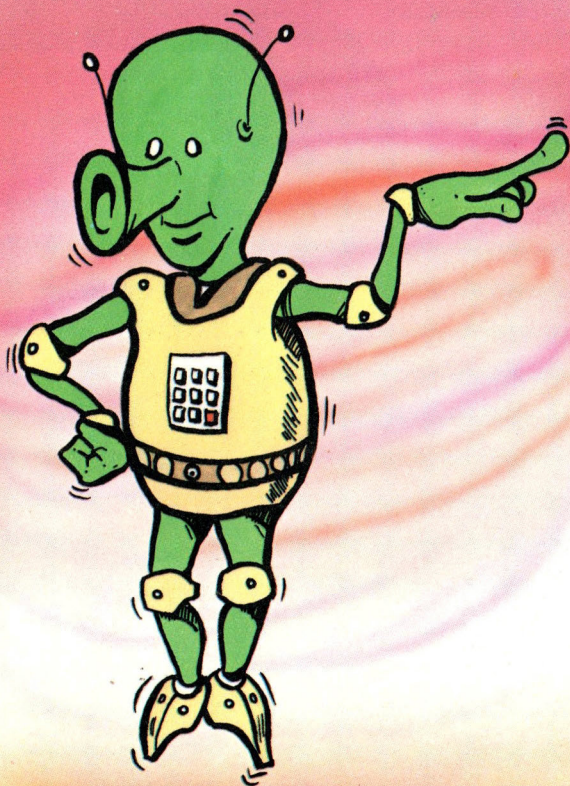
TAB (espressione)

dove espressione è un valore numerico compreso tra 0 e 65.535, anche se in pratica è tra 0 e 31.

LINGUAGGIO

L'errore è sempre in attesa: malignamente aspetta una tua distrazione, anche minima.
Non dargli mai la possibilità di sogghignare.
Impara la sintassi delle istruzioni.
Realizza programmi in forma modulare, facilmente leggibili, e molto documentati.
Nella digitazione, poi, metti la massima attenzione: anche una virgola al posto sbagliato, compromette la buona riuscita del programma.

Uso avanzato di PRINT ed INPUT

Imparerai adesso a fare tesoro delle varie istruzioni di INGRESSO/USCITA che hai imparato. Parleremo infatti dell'uso avanzato di PRINT ed INPUT. La parola «avanzato» lascia intuire numerose possibilità, e di fatti essa indica tutte le regole, i segreti ed i trucchi che permettono al programmatore di ottenere pieno ed assoluto controllo di

tutto ciò che entra od esce dal calcolatore.
Un primo, utile comando può essere

```
POKE 23692,255
```

Esso permette di evitare che la macchina, dopo aver riempito di lettere o numeri lo schermo, scriva sempre SCROLL? alla fine del video, arrestando di conseguenza l'esecuzione.
Per esempio, prova:

```
10 FOR X=65 TO 127
20 PRINT X; TAB 8; CHR$(X)
30 POKE 23692, 255
40 NEXT X
50 IF INKEY$< >“ ” GO TO 10
```

Scoprirai che la visualizzazione dei caratteri creati all'interno del ciclo FOR dall'istruzione CHR$ non subirà alcun arresto né pausa di esecuzione.
L'unico modo per arrestare il programma (linea 50) sarà quello di premere un tasto qualsiasi.
Per ristabilire la normale procedura di scroll, batti:

```
POKE 23692,1
```

Vediamo ora qualcosa su INPUT: questo comando è infatti in grado di fare qualcosa di più di quanto tu non abbia visto finora.
Supponi di voler scrivere un programma per imparare a fare mentalmente le moltiplicazioni.
Desideri che in ingresso ti venga richiesta una coppia di numeri qualsiasi e che in seguito compaia una domanda alla quale dovrai rispondere con il risultato della moltiplicazione.
Il problema però è questo: i numeri in ingresso verranno assegnati a due variabili (per esempio A e B).
Come sarà possibile assegnare ad una terza variabile (chiamiamola C) il numero fornito come risultato, mediante una istruzione INPUT?
Una linea di programma come

```
INPUT “QUANTO FA”;
 A; “*”; B; “?”; C
```

considererebbe infatti tutti gli elementi al di fuori delle virgolette come variabili da assegnare.

La difficoltà (così lunga da spiegare!) è invece di breve e semplice superamento: basterà racchiudere entro parentesi i nomi delle variabili:

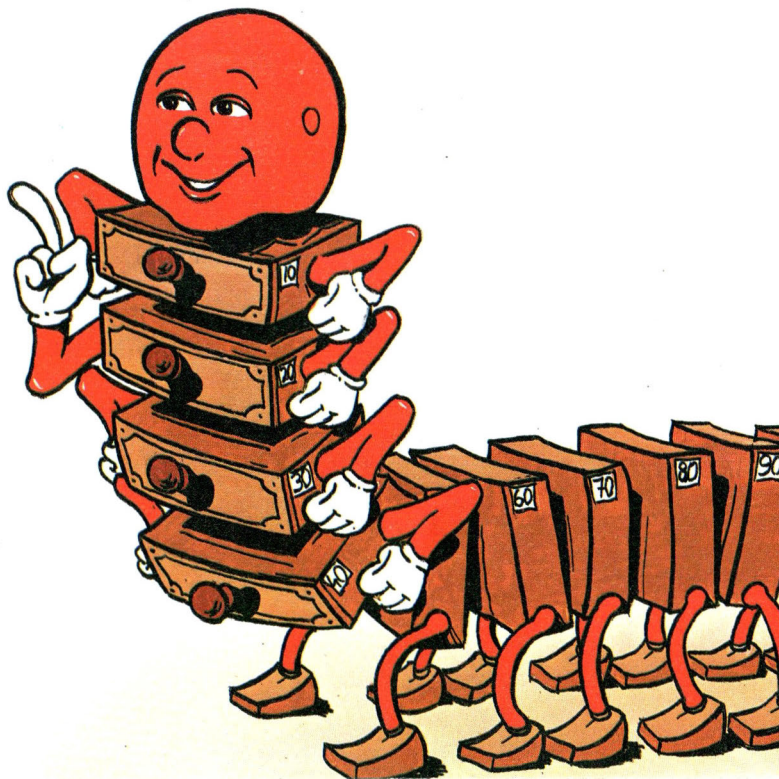
```
INPUT "QUANTO FA"; (A); "*"; (B); "?"; C
```

Tutto risolto. Più semplice di così! Questa regola è generale. Qualsiasi espressione che inizi con una lettera, e che deve essere stampata come parte di un messaggio, deve essere racchiusa tra parentesi. Il programma completo potrà quindi essere una cosa del genere:

```
 10 LET RIGA=10
 20 CLS: INPUT AT 10, 0; "QUAL È IL PRIMO NUMERO?   "; A
 30 INPUT AT 12, 0; "QUAL È IL SECONDO NUMERO?   "; B
 40 CLS: INPUT AT RIGA, 0; "QUANTO FA"; (A); "*"; (B); "?"; C
 50 IF C=A*B THEN PRINT AT RIGA+4,0; "BRAVO!!": GO TO 100
 60 PRINT "SBAGLIATO!!";
 70 LET RIGA=RIGA+2
 80 IF RIGA <=14 THEN PRINT TAB 15; "RIPROVA.": PAUSE 100:
    GO TO 40
 90 PRINT AT 21,0;"IL RISULTATO DI"; A;"*";B;"È"; A*B
100 INPUT "VUOI GIOCARE ANCORA?   "; R$ : IF R$ = "S" THEN RUN
110 REM FINE
```

Il listato non richiede molti commenti: l'unica particolarità degna di rilievo appare alla riga 80.
La linea è infatti costituita da più istruzioni (regolarmente separate dai due punti), la prima delle quali è una IF. Devi fare attenzione al fatto che se la condizione della IF è falsa (cioè non è verificata), in quella riga

non viene eseguita nessuna delle istruzioni: l'interprete BASIC salta infatti l'intera linea, portando l'esecuzione alla istruzione seguente (cioè alla linea 90).
Ancora una cosa, stavolta sia su PRINT che su INPUT.
Capita talvolta di voler stampare sullo schermo qualche parola racchiusa entro virgolette. Ciò non è però possibile in condizioni normali. Le «virgolette» ("") nel BASIC sono infatti utilizzate come delimitatori: tutte le volte che l'interprete BASIC ne incontra una, la sottintende come carattere iniziale o finale di una costante di tipo stringa.
Come fare allora? La risposta si chiama codice ASCII.
Sarà cioè sufficiente utilizzare il codice ASCII delle virgolette (che è 34) per «scavalcare» l'interprete BASIC.
Immaginiamo per esempio di voler scrivere sullo schermo la frase:
"Tutto è bene ciò che finisce bene"
L'istruzione per visualizzarla sul video racchiusa tra virgolette dovrà allora essere:

```
PRINT CHR$(34); "TUTTO È BENE CIO' CHE FINISCE BENE"; CHR$(34)
```

VIDEOESERCIZI

Usando la funzione TAB, ricerca un nuovo metodo per stampare la tabellina di un numero, in modo che le unità, le decine ecc. dei vari prodotti risultino perfettamente incolonnate. Ecco, per aiutarti, un esempio con quella del 7.

```
10 LET C = 10 : LET S = C : REM LE UNITA' VANNO STAMPATE ALLA COLONNA 10
20 CLS
30 FOR V = 1 TO 10
40 LET P = V * 7
50 IFP > 9 THEN LET S = C - 1
60 PRINT TAB S; P
70 NEXT V
```

Prova a sostituire la linea 30 con : 30 FOR V = 1 TO 20.
Suggerimento: per incolonnare le centinaia, introduci tra la linea 50 e la 60 un ulteriore controllo.

Gioca a dadi con il tuo Spectrum

```
 10 CLS
 20 INPUT "GIOCATORE 1 ="; G$
 30 INPUT "GIOCATORE 2 ="; H$
 40 PRINT AT 1, 10; "DADOMATTO" ' ' ' '
 50 PRINT G$; " PREMI UN TASTO"
 60 PAUSE 0
 70 LET R1 = INT (6 * RND) + 1
 80 PRINT TAB 10; R1; ' '
 90 PRINT H$; " PREMI UN TASTO"
100 PAUSE 0
110 LET R2 = INT (6 * RND) + 1
120 PRINT TAB 10; R2
130 IF R1 = R2 THEN PRINT ' ' ; TAB 4; "PARTITA PARI" : GO TO 200
140 IF R1 > R2 THEN PRINT ' ' ; TAB 4; "VINCE "; G$ : GO TO 200
150 PRINT ' ' ; TAB 4; "VINCE "; H$
200 PRINT ' ' ; "ANCORA ? S/N"
210 PAUSE 0 : LET T$ = INKEY$
220 IF T$ = "S" OR T$ = "s" THEN CLS : GO TO 40
230 CLS : PRINT AT 12, 14; "FINE"
```

Modifica le linee 70 e 110 in modo da simulare il lancio di 2 dadi: pensa a quali numeri, minimo e massimo possono uscire.